International Journal of Electrical and Electronics
Engineering Research (IJEEER)
ISSN(P): 2250-155X; ISSN(E): 2278-943X
Vol. 3, Issue 5, Dec 2013, 11-18

BCD ADDER AND MULTIPLIER USING REVERSIBLE LOGIC DESIGN

M SAI KRISHNA SHANKAR & B VISHNU MOHAN

Department of Electronics and Communication Engineering, Osmania University, Hyderabad, Andhra Pradesh, India

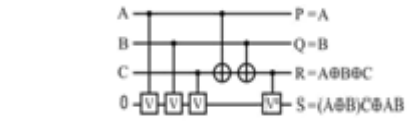
ABSTRACT

Filters are widely used in the world of communication and computation. To design a finite impulse response (FIR) filter that satisfies all the required conditions is a challenge. Power consumption by the multiplier and adder blocks in the architecture is the prime cause for concern in FIR design. In this paper, design of an FIR filter entirely using Reversible logic is presented. Reversible logic is emerging as a promising computing paradigm having widespread areas of application. It is based on the fundamental that zero energy dissipation would be achieved if the design is made of reversible gates. Furthermore, majority of the design is made of Peres gate which has the least quantum cost of all the existing reversible gates. Thus, this paper presents the first and novel approach in realizing an FIR filter that can overcome the problem of power dissipation. The first and foremost approach in designing a low pass FIR filter entirely using reversible logic is presented in this paper. The design is the most novel for it, dominantly makes use of a Peres Gate (quantum cost = 4) in the realization of the multiplier and adder blocks which form major components of the FIR filter block. Power dissipation, the most vital issue in the design of a circuit, can be resolved to the maximum extent by following the proposed implementation.

KEYWORDS: Communication and Computation, FIR Filter

INTRODUCTION

Among the emerging computing paradigms, reversible logic appears to be promising due to its wide applications in various technologies. Reversible circuits are those circuits that do not lose information during computation and reversible computation in a system can be performed only when the system comprises of reversible gates. These circuits can generate unique output vector from each input vector, and vice versa, that is, there is a one-to-one mapping between the input and the output vectors. Some of the emerging nanotechnologies having applications of reversible logic are quantum computing, quantum dot cellular automata, optical computing, Spintronics, DNA computing, molecular computing and also in power-efficient nano-computing, etc



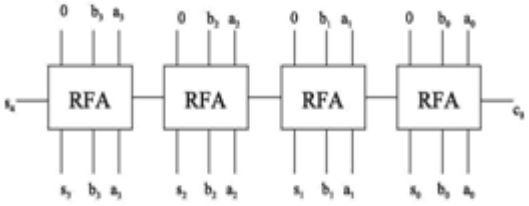


Figure 1(a): 1 Bit Reversible Adder
Figure 1 (b): 4 Bit Reversible Adder Using Conventional Ripple Carry Approach

For example, the design of 1 bit reversible full adder having inputs A, B and carry input C is shown in Figure 1(a)

which shows that the design of a 1 bit reversible full adder needs 1 ancilla input having constant value as 0. The sum and carry outputs are produced at outputs R and S, respectively (the regenerated inputs are not considered as garbage outputs). Thus, as shown in Fig.1 (b) for a 4 bit reversible adder, the design of an n bit reversible adder based on the conventional ripple carry approach of cascading will need n ancilla inputs. This shows that the use of the conventional approach of design and synthesis to reversible logic circuits may result in significant overhead in terms of parameters such as ancilla and garbage bits. Further, in reversible logic based circuit design, parameters such as ancilla inputs, garbage outputs, quantum cost and delay need to be optimized that are completely different from the traditional parameters of speed, power and chip area used in conventional com-puting. Thus, there is a need of research towards developing new design and synthesis methods for realization of reversible arithmetic circuits and a synthesis framework in which multiple parameters can be optimized.

In summary, the motivations for this dissertation are: (i) to explore the design and synthesis of reversible logic circuits considering metrics of ancilla inputs, garbage outputs, quantum cost and the delay, and (ii) to explore the benefits of reversible logic towards concurrent and offline testing of faults in circuits based on emerging nanotechnologies.

Contributions and Significance

In reversible logic design and synthesis there are a few important points that need to be kept in mind for efficient and optimal reversible design

- Minimize the quantum cost of the design. The quantum cost of a design is the number of 1x1 and 2x2 reversible gates used in its design.
    - Minimize garbage outputs by using as many outputs of every gate as possible.
    - Minimize delay by reducing the logic depth.
    - Avoid constant inputs to gates unless necessary. The constant inputs are known as ancilla inputs.
    - Avoid a fan-out of more than one, as each fan-out greater than one requires an additional copying circuit (fan-out is not allowed in reversible logic).

PROPOSED WORK ON THE IMPLEMENTATION OF FIR FILTER USING REVERSIBLE LOGIC

FIR Filter

Before giving a detailed insight of the work carried out in the designing of the FIR filter using reversible logic, let us once again briefly go through what a FIR filter is:

A FIR filter is one whose output *y* of a linear time invariant system is determined by convolving its input signal *x* with its impulse response *b*.

For a discrete-time FIR filter, the output is a weighted sum of the current and a finite number of previous values of the input. The operation is described by the following equation, which defines the output sequence *y[n]* in terms ofsequence *x[n]*:

$$\begin{aligned} y[n] &= b_0 x[n] + b_1 x[n-1] + \cdots + b_N x[n-N] \\ &= \sum_{i=0}^{N} b_i x[n-i] \end{aligned}$$

Where:

- $x[n]$ is the input signal,
- $y[n]$ is the output signal,
- b_i are the *filter coefficients*, also known as *tap weights*, that make up the impulse response,
- N is the filter order.

Working of an FIR Filter

Having understood that, let us proceed to understand the working or functioning of a FIR filter:

FIR filters are often used because they are simple and easy to understand .from the above figure, one can understand that, an FIR filter works by multiplying an array of the most recent n data samples by an array of constants (called the tap coefficients), and summing the elements of the resulting array. (This operation is commonly called a dot product.) The filter then inputs another sample of data (which causes the oldest piece of data to be thrown away) and repeats the process.

An FIR filter can be easily implemented using just three digital hardware elements, a unit delay (a **latch**), a multiplier, and an adder. The unit delay simply updates its output once per sample period, using the value of the input as its new output value. In the convolution sum, notice that at each *n* we need access to $x(n)$, $x(n-1)$, $x(n-2)$, …, $x(n-M)$.

We can maintain this set of values by cascading a set of latches to form a **delay line**, as shown below:

$$y(n) = \sum_{m=0}^{M} h(m)\, x(n-m)$$

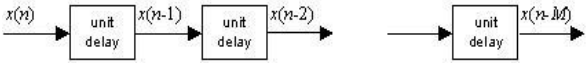


Figure 2(a): Delay Line

For each integer *n*, the output sample is the values in the delay line scaled by $h(0)$, $h(1)$,

…, $h(M)$. To obtain these values, we simply **tap** the delay line, as shown in the following figure

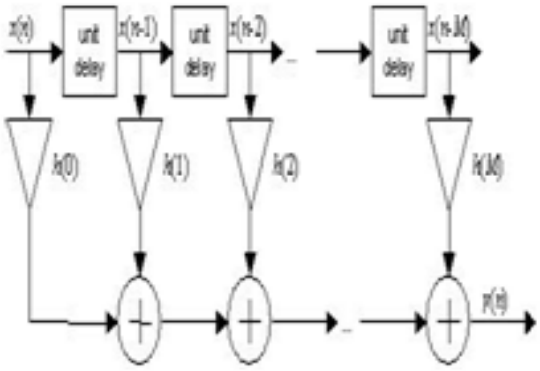


Figure 2(b): Tapped Delay Line FIR Filter

The triangular boxes denote multipliers that multiply by a constant ($h(m)$). The circles denote adders. The above picture shows a **tapped delay line** implementation of an FIR filter.

Hence, it follows that the operation of filtering of a signal $x(m)$ can be mathematically expressed as the convolution of the input signal and the impulse response of the filter $h(m)$ as

$$y(m) = \sum_{k=1}^{N} h(k)\, x(m-k)$$

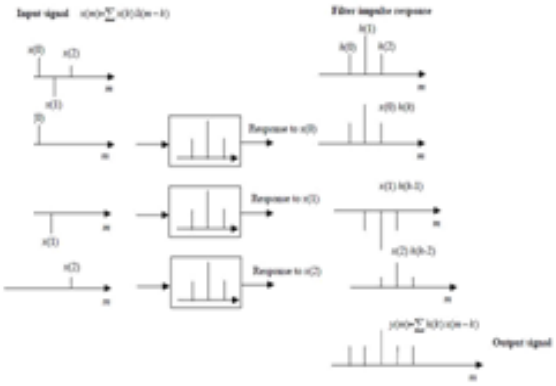


Figure 2(c): Illustration of the Output of a Filter in Response to an Input as the Sum of Impulse Responses of the Filter to Individual Input Samples

This is the convolution of input and the impulse response.

PROPOSED ARCHITECTURE

Stage One

Stage one is the beginning of the work with a first order FIR filter.

Figure 3 (a) shows the conventional architecture of the

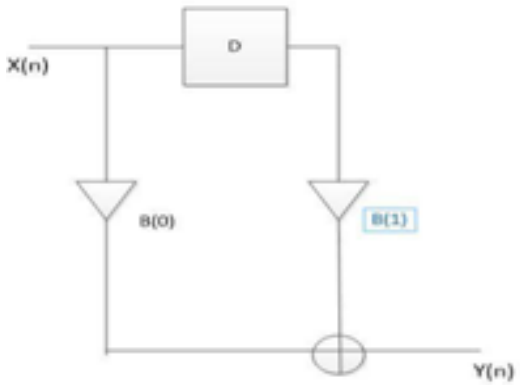


Figure 3(a): Conventional Architecture of Stage One

From the figure 3 (a), it is evident that a delay, multiplier and adder blocks are required.

The following figures show the design of each of these blocks using reversible logic gates

Delay Block

Each delay element in the delay block is designed using a Fredkin and a Feynman gate. The characteristic equation of the D latch can be written as

Q+ = D.E + E˝Q here Q+ is the current output D is the input

E is the Enable or the Clock signal Q is the previous output

Multiplier

Multiplier performs the most important part of the FIR implementation where it multiplies the input with the filter coefficients.The proposed design makes use of a Peres gate to perform the multiplication of the coefficient and the input values.

A one bit multiplier is sufficient for the implementation of a 1st order filter.

Figure 3(b): Shows that when the Third Input of A Peres Gate is Taken as Zero, the Third Output Yields Product of the First Two Terms

Adder

The adder sums up the outputs from the filters to give the final filtered output.

A Peres gate also functions as an adder in this case for its second output yields the sum of the first two inputs when the third input is maintained a constant zero.

Figure 3(c): Adder Block of First Order FIR Filter

SIMULATION RESULTS

As we have seen above, our work comprises of the implementation of a BCD Adder using four different detectors, the implementation of the multiplier and the work on the filter i.e., first order low-pass FIR filter using reversible logic. In this chapter we provide the simulation results of individual blocks of the filters along with the final simulation result.

BCD ADDER

Simulation Results of Ripple Carry Adder

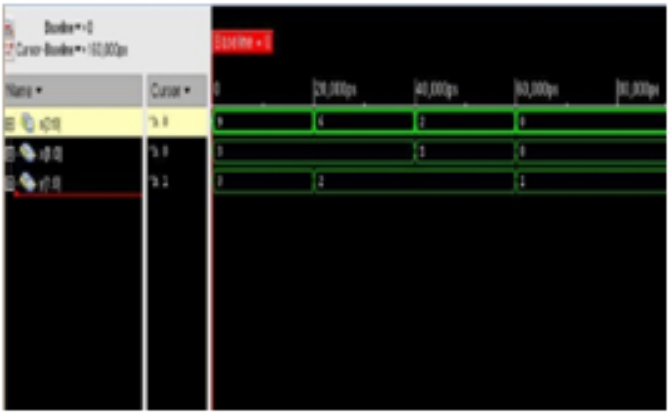


Figure 4

Simulation Results of Detectors with Different Gates

Figure 5

Simulation Results of Final BCD Adder

Figure 6

Simulation Results of First Order FIR Filter Using Reversible Logic

Delay Block

The Delay block is programmed using Verilog codes in Cadence digital labs and the simulation result for first order FIR filter is shown in the Figure.

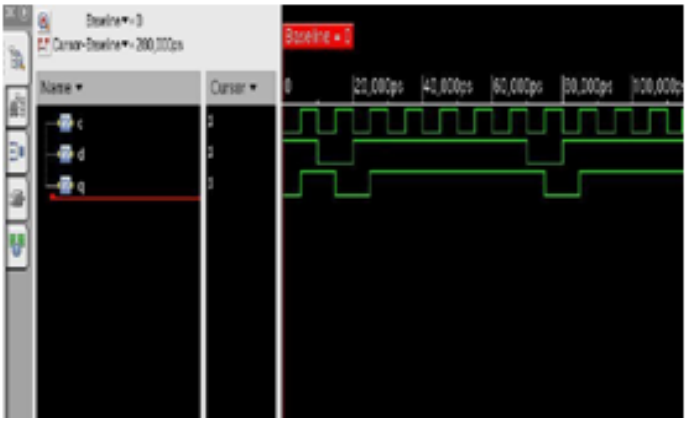


Figure 7

Here c is the clock, d is the input and q is the delayed output.

Multiplier Block

The multiplier block used in first order FIR filter is programmed and compiled in Verilog code using Cadence digital labs. The simulation result of the multiplier block is shown in the Figure.

Figure 8

Adder Block

The adder block is designed and programmed in Verilog code using Cadence Digital lab tools. The simulation result of the adder block is shown in the Figure.

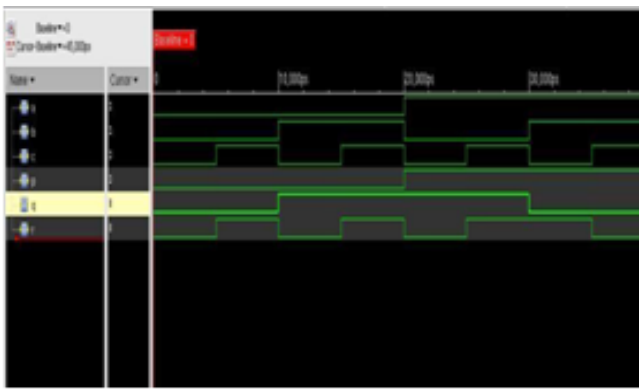


Figure 9

Final Simulation Result of First Order FIR Filter

Collaborating all the blocks we have the final simulation output of the first order FIR filter architecture. The simulation result is as shown in the Figure.

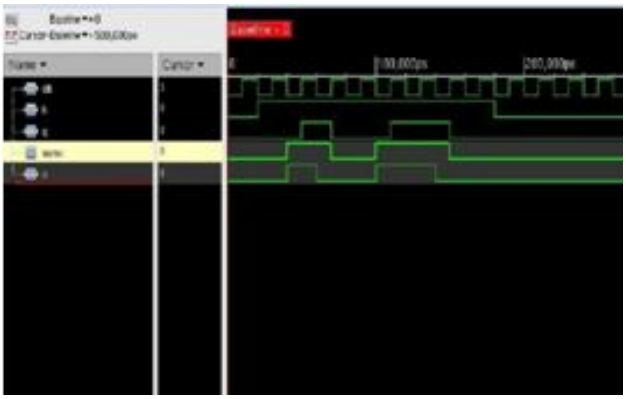


Figure 10

CONCLUSIONS AND FUTURE SCOPE

Conclusions

For the design of FIR filter using Reversible logic, the different architectures of the delay, adding and multiplying blocks have been collaborated and simulated.

The first and foremost approach in designing a low pass FIR filter entirely using reversible logic is presented in this thesis. The design is the most novel for it, dominantly makes use of a Peres Gate (quantum cost = 4) in the realization of the multiplier and adder blocks which form major components of the FIR filterblock. Power dissipation, the most vital issue in the design of a circuit, can be resolved to the maximum extent by following the proposed implementation.

Future Scope

The work done in this project can be extended to the implementation of designing a High-pass FIR filter using reversible logic. It can also be extended design the filter using different windowing techniques and the design order can be experimented if required.

REFERENCES

1. *H.Thapliyal*. Design, synthesis and test of reversible logic circuits for emerging nanotechnologies.Ph.D dissertation, univ. of South Florida,Tampa. 2011. [online]. Available:http://scholarcommons.usf.edu/etd/3379
2. A.J.Genot ,J.Bath and A.J. Turberfield Reversible Logic Circuits Made of DNA, Journal of the American Chemical Society; Nov 2011.
3. R. Zhang, K.Walus, W. Wang and G.A. Jullien. A method of majority logic reduction for quantum cellular automata, IEEE Trans. Nanotechnology, 3(4):443-450, December 2004.
4. S.Bandyopadhyay and M.Cahay Introduction to Spintronics, CRC press, Boca Raton,2008